LCNS, Vol 2932, pp. 349–360, Springer 2004

Avoiding Forbidden Submatrices by Row Deletions[*]

Sebastian Wernicke, Jochen Alber, Jens Gramm, Jiong Guo, and
Rolf Niedermeier

Wilhelm-Schickard-Institut für Informatik, Universität Tübingen,
Sand 13, D-72076 Tübingen, Fed. Rep. of Germany
{wernicke,alber,gramm,guo,niedermr}@informatik.uni-tuebingen.de

Abstract. We initiate a systematic study of the Row Deletion(B) problem on matrices: For a fixed "forbidden submatrix" B, the question is, given an input matrix A (both A and B have entries chosen from a finite-size alphabet), to remove a minimum number of rows such that A has no submatrix which is equivalent to a row or column permutation of B. An application of this question can be found, e.g., in the construction of perfect phylogenies. Establishing a strong connection to variants of the NP-complete Hitting Set problem, we show that for most matrices B Row Deletion(B) is NP-complete. On the positive side, the relation with Hitting Set problems yields constant-factor approximation algorithms and fixed-parameter tractability results.

1 Introduction

Forbidden subgraph problems play an important role in graph theory and algorithms (cf., e.g., [1, Chapter 7]). For instance, in an application concerned with graph-modeled clustering of biological data [9] one is interested in modifying a given graph by as few edge deletions as possible such that the resulting graph consists of a disjoint union of cliques (a so-called *cluster graph*). Exact (fixed-parameter) algorithms to solve this NP-complete problem make use of the fact that a graph is a cluster graph iff it contains no vertex-induced path of three vertices as a subgraph [3, 4]. There is a rich literature dealing with such "graph modification problems," cf., e.g., [6]—many problems here being NP-complete.

By way of contrast, in this paper we start the so far seemingly widely neglected investigation of forbidden submatrix problems from an algorithmic point of view. Here, given an input matrix A and a fixed matrix B, the basic question is whether B is *induced* by A. This means that a *permutation* B' of B—that is, B can be transformed into B' by a finite series of row and column swappings—can be obtained from A by row and column deletions. This work studies corresponding "matrix modification problems" where, given A and a fixed B, we are asked

[*] Supported by the Deutsche Forschungsgemeinschaft (DFG), project PEAL (parameterized complexity and exact algorithms), NI 369/1; project OPAL (optimal solutions for hard problems in computational biology), NI 369/2; junior research group "PIAF" (fixed-parameter algorithms), NI 369/4.

to remove as few rows from A as possible such that the resulting matrix no longer induces B. Forbidden submatrix problems, e.g., are motivated by questions of computational biology concerning the construction of "perfect phylogenies" [8, 10]. Here, a binary input matrix A allows for a perfect phylogeny iff A does not induce the submatrix B consisting of the rows $(1, 1)$, $(1, 0)$, and $(0, 1)$ (see [8, 10] for details).

We initiate a systematic study of matrix modification problems concerning the complexity of row deletion for forbidden submatrices. Our main result is to establish a very close link between many of these problems and restricted versions of the NP-complete Hitting Set problem [2]. We describe and analyze structures of the forbidden submatrix B which make the corresponding row deletion problem "equivalent" to particular versions of Hitting Set. On the negative side, this implies NP-completeness for most row deletion matrix modification problems, holding already for binary alphabet. On the positive side, we can also show that approximation and fixed-parameter tractability results for Hitting Set carry over to the corresponding row deletion problems. To the best of our knowledge, no such systematic study has been undertaken so far. We are only aware of the related work of Klinz et al. [5] dealing with the permutation of matrices (without considering row deletions) in order to avoid forbidden submatrices. There, however, they consider the case of permuting rows and columns of the "big" matrix A to obtain a matrix A' such that A' cannot be transformed into a fixed matrix B by row and column deletions. Among other things, they show NP-completeness for the general decision problem.

The paper is structured as follows. In Section 2, we start with the basic definitions and some easy observations. After that, in Section 3, the main results of the work are presented, giving (or sketching) several "parameter-preserving reductions" (the core tool of this paper) from Hitting Set problems to Row Deletion(B) for various types of the forbidden submatrix B. Then, in Section 4 we show how Row Deletion(B) can be solved using algorithms for Hitting Set problems, again using a parameter-preserving reduction. Finally, we end with some concluding remarks and open problems in Section 5.

Due to the lack of space, several proofs have been omitted or shortened in this article, more details can be found in [11].

2 Definitions and Preliminaries

All matrices in this work have entries from an alphabet Σ of fixed size ℓ; we call these matrices *ℓ-ary*. Note, however, that all computational hardness results already hold for binary alphabet. The central problem Row Deletion(B) for a fixed matrix B is defined as follows.

Input: A matrix A and a nonnegative integer k.
Question: Using at most k row deletions, can A be transformed into a matrix A' such that A' does not induce B?

Herein, A' *induces* B if there exists a B' obtained from B through a finite series of row and column swappings such that B' can be obtained from A by row and column removal. The remaining rows and columns of A resulting in B' are called *occurrence* of B in A. A matrix A is *B-free* if B is not induced by A.

This work establishes strong links between Row Deletion(B) and the d-Hitting Set problem for constant d, which is defined as follows.

> **Input:** A collection $\mathcal{C}$ of subsets of size at most d of a finite set $\mathcal{S}$ and a nonnegative integer k.
> **Question:** Is there a subset $S' \subseteq \mathcal{S}$ with $|S'| \leq k$ such that S' contains at least one element from each subset in $\mathcal{C}$?

Already for $d = 2$, d-Hitting Set is NP-complete [2].

To express the closeness between variants of Row Deletion(B) and d-Hitting Set for various d, we need the following strong notion of reducibility. Let $(\mathcal{S}, \mathcal{C}, k)$ be an instance of d-Hitting Set. We say that there is a *parameter-preserving reduction* from d-Hitting Set to Row Deletion(B) if there is a polynomial time algorithm that transforms $(\mathcal{S}, \mathcal{C})$ into a matrix A and $(\mathcal{S}, \mathcal{C}, k)$ is a true-instance of d-Hitting Set iff (A, k) is a true-instance of Row Deletion(B). The important observation here is that the "objective value parameter" k remains unchanged. This makes it possible to link approximation and exact (fixed-parameter) algorithms for both problems.

Finally, for actually performing row deletions in the input matrix A of Row Deletion(B), it is necessary to find the set of rows in A that induce B. A straightforward algorithm yields the following.

Proposition 1 *Given an $n \times m$ matrix A and a fixed $r \times s$ matrix B (where $1 \leq r \leq n$ and $1 \leq s \leq m$), we can find all size-r sets of rows in A that induce B in $O(n^r \cdot m \cdot s \cdot r!)$ worst-case time.* □

Observe that Proposition 1 gives a pure worst-case estimation not making use, e.g., of the special structure of the respective B. In any case, however, for constant values of r and s the running time is polynomial.

3 Computational Hardness

In this section, we explore the relative computational hardness of Row Deletion(B) by studying its relationship to d-Hitting Set. We point out many cases concerning the structure of B for which Row Deletion(B) is at least as hard as d-Hitting Set for some d depending only on B.

Summary of results.
The key idea behind all reductions from d-Hitting Set to Row Deletion(B) is to choose a symbol σ from the alphabet Σ and to decompose B—in a certain manner—into four submatrices, one of them consisting only of σ's and one of them which does not contain σ. We can then use the latter submatrix to encode a given d-Hitting Set instance into an instance of Row Deletion(B) (i.e., a matrix A) and use σ as a "fill-in" symbol to prevent unwanted occurrences of B in A.

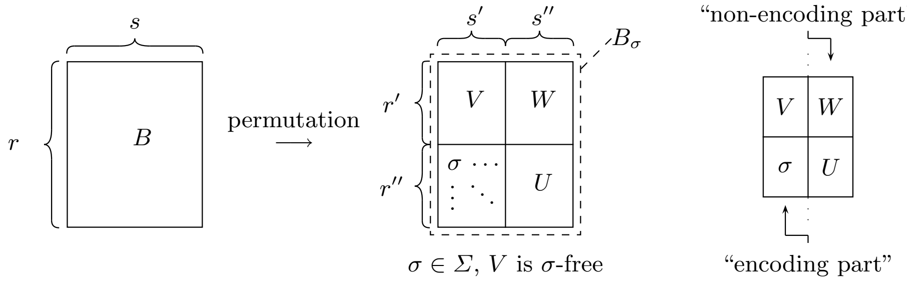


Fig. 1. General scheme for the σ-decomposition of a matrix B over the alphabet σ.

We call this special decomposition of the forbidden submatrix B a *σ-decomposition*, illustrated in Fig. 1 and formally defined as follows:

Definition 1. (σ-DECOMPOSITION) *Given an ℓ-ary $r \times s$ matrix $B = (b_{ij})$ over the alphabet Σ. A permutation B_σ of B is called a* σ-decomposition *of B if there exists a $\sigma \in \Sigma$ and there exist r', r'', s', s'' with $r' + r'' = r, s' + s'' = s$ such that*
(1) $r' > 0$ and $s' > 0$,
(2) $\forall\ 1 \le i \le r', 1 \le j \le s' :\ b_{ij} \neq \sigma$ (call this upper left submatrix V) and
(3) $\forall\ r' < i \le r, 1 \le j \le s' :\ b_{ij} = \sigma$.
The upper right $r' \times s''$ submatrix $(b_{ij})_{1 \le i \le r', s' < j \le s}$ of B_σ is called W, the lower right $r'' \times s''$ submatrix $(b_{ij})_{r' < i \le r, s' < j \le s}$ is referred to as U.

The left part $(b_{ij})_{1 \le i \le r, 1 \le j \le s'}$ of B_σ (the one containing V) is called the encoding part *of B_σ. The right part $(b_{ij})_{1 \le i \le r, s' < j \le s}$ of B_σ (the one consisting of W and U) is called the* non-encoding part *of B_σ.*

For a given $\sigma \in \Sigma$, a corresponding σ-decomposition can be easily computed in time linearly depending on the size of B (details omitted). In the following hardness proofs, the height of V plays a crucial role.

Our main hardness results then are the following.

Theorem 1. *Let B be a forbidden submatrix of size $r \times s$ with a σ-decomposition B_σ where the submatrix V (of height r') of B_σ is not induced in the non-encoding part of B_σ. Then there exists a parameter-preserving reduction from r'-*Hitting Set *to* Row Deletion*(B). Hence, if $r' \ge 2$,* Row Deletion*(B) is NP-complete.*

Clearly, it is possible that V *is* induced in the non-encoding part of B_σ. In particular, then each column vector of V is induced at least once in the non-encoding part. If we can find one column vector of V which is induced *at most once* in the non-encoding part, we again are able to achieve a hardness result for Row Deletion(B):[1]

[1] Observe that it is possible to construct a submatrix B which fulfills the prerequisites of Theorem 1, but does not fulfill the prerequisites of Theorem 2, and vice versa.

Theorem 2. *If the $r \times s$ submatrix B has a σ-decomposition B_σ where the submatrix V of height r' has a column vector v that is induced at most once in the non-encoding part of B_σ, then r'-*Hitting Set *is parameter-preserving reducible to* Row Deletion*(B).*

If the submatrix B does not fulfill any of the two prerequisites from Theorems 1 or 2, we can determine two further subcases for which a hardness result can be established:

Theorem 3. *Let B be a forbidden $r \times s$ submatrix with a σ-decomposition B_σ where all entries of U are equal to σ and V contains r' rows. Then r'-*Hitting Set *is parameter-preserving reducible to* Row Deletion*(B).*

Theorem 4. *Let B be a forbidden $r \times s$ submatrix with a σ-decomposition B_σ where all entries of W are equal to σ and V contains r' rows. Then r'-*Hitting Set *is parameter-preserving reducible to* Row Deletion*(B).*

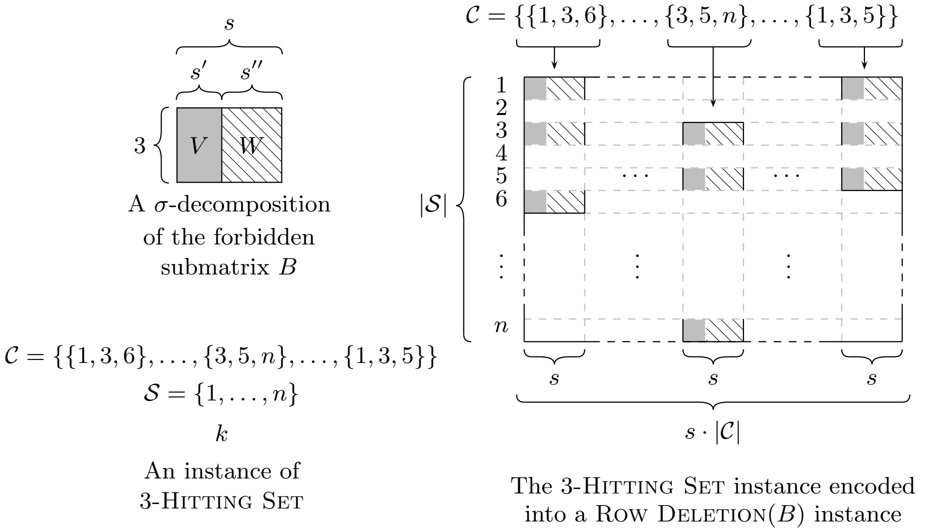
For all other cases, i.e., if B does not fulfill any of the prerequisites from Theorems 1–4, we are not aware of a general statement on the complexity of Row Deletion(B). As an example of such a matrix consider $B = \left(\begin{smallmatrix} 1&0&0\\ 0&1&0\\ 0&0&1 \end{smallmatrix}\right)$ over the alphabet $\Sigma = \{0,1\}$.

It is also clear that Row Deletion(B) is not NP-hard for all B. For example, Row Deletion(B) is solvable in polynomial time if B is a 1×1-matrix. Besides, there are also non-trivial examples for which Row Deletion(B) is solvable in polynomial time, as the matrix $B = \left(\begin{smallmatrix} 1\\ 0 \end{smallmatrix}\right)$ over the alphabet $\Sigma = \{0,1\}$ shows: Observe that a B-free matrix A has the property that all its columns consist either solely of 1's or solely of 0's, i.e., all rows of A are identical. This implies that the minimum number of rows that need to be deleted in order to make an $n \times m$ matrix B-free is equal to $n - x$, where x denotes the size of the largest set of identical rows in A, which can be determined efficiently.

Observe, however, that Row Deletion(B) for $B = \left(\begin{smallmatrix} 1\\ 1 \end{smallmatrix}\right)$ over the alphabet $\Sigma = \{0,1\}$ already is NP-complete by Theorem 1, since B trivially has a σ-decomposition with $\sigma = 0$, $V = B$, and $W = U = \emptyset$.

Outline of hardness proofs.
As mentioned above, all hardness results are proven by encoding an instance $(\mathcal{S}, \mathcal{C}, k)$ of d-Hitting Set—where the value of d is determined by the forbidden submatrix B—as an instance (A, k) of Row Deletion(B). The key idea concerning how to use a given σ-decomposition of B to encode a d-Hitting Set instance is illustrated by the following proof of Theorem 3. Subsequently, we indicate how this construction can be extended to prove, in ascending involvedness of the construction, Theorem 4, Theorem 1, and Theorem 2. Note that due to lack of space and for a better understandability of the key ideas involved, parts of some proofs are only sketched; for their details, refer to [11].

Proof (Theorem 3). Given an instance $(\mathcal{S}, \mathcal{C}, k)$ of r'-Hitting Set and a σ-decomposition B_σ of the forbidden $r \times s$ submatrix B. Let $\mathcal{S} = \{1, \ldots, n\}$ and



s
s'
s''
3
V
W
$\mathcal{C} = \{\{1,3,6\},\ldots,\{3,5,n\},\ldots,\{1,3,5\}\}$
$\mathcal{S} = \{1,\ldots,n\}$
k
$\mathcal{C} = \{\{1,3,6\},\ldots,\{3,5,n\},\ldots,\{1,3,5\}\}$
1
2
3
4
5
6
n
$|\mathcal{S}|$
s
s
s
$s \cdot |\mathcal{C}|$

A σ-decomposition of the forbidden submatrix B

An instance of 3-Hitting Set

The 3-Hitting Set instance encoded into a Row Deletion(B) instance

Fig. 2. An example reduction from 3-Hitting Set to Row Deletion(B) following from the proof of Theorem 3 (illustrated for the case where $r' = r$).

$\mathcal{C} = \{C_1, \ldots, C_m\}$. For now, assume that $r' = r \Rightarrow r'' = 0$ (i.e., B consists only of V and W). We generate a matrix A of size $n \times (s \cdot m)$. Each row of A corresponds to an element in $\mathcal{S}$. For each set $C \in \mathcal{C}$, one occurrence of B is encoded into s consecutive columns of A, using the rows that correspond to the elements in C. For example, consider $C_h \in \mathcal{C}$, $1 \leq h \leq m$, with $C_h = \{z_1, \ldots, z_r\}$ and $z_1, \ldots, z_r \in \{1, \ldots, n\}$. Then, we generate the submatrix $(a_{ij})_{1 \leq i \leq n, (h-1)\cdot s < j \leq h \cdot s}$ of A such that the z_ith row of this submatrix equals the ith row of B, for all $i = 1, \ldots, r$, and all other rows of this submatrix are set equal to σ (an illustration for this is provided in Fig. 2). In this way, A contains m blocks of s consecutive columns where each block induces B exactly once. These are the only occurrences of B, since two columns from two different blocks cannot have r rows containing no σ. Observe how the property of A that B is only induced within each block and, furthermore, exactly once in each block, is due to using σ as a "fill-in" symbol: Since σ is not contained in V, we ensure that, by using σ as the "default-entry" for A, no additional occurrences of V (and therefore B) are induced in the rows of a block other than the ones intended. The outlined reduction can be performed in $O(n \cdot m \cdot s)$ time.

The construction is easily generalized for the case where $r' < r$, i.e., $r'' > 0$, by adding $r'' + k$ rows containing only σ at the bottom of A. Note that by this construction, although B is induced multiple times in each block, V is induced only once in each block.

It can easily be shown that solutions to the original instance of r'-Hitting Set have a "1:1-correspondence" with solutions to Row Deletion(B) on A:

"$\Rightarrow$" Assume that we have a solution $\mathcal{S}'$ to the original instance of r'-Hitting Set with $|S'| = k$. Then, delete those rows in A that correspond to the elements in $\mathcal{S}'$, thus obtaining A'. Note that then, from the submatrix V of each B that was encoded into A, at least one row has been deleted. Every column in A' contains less than r' symbols different from σ. This directly implies that V does not occur in A', and therefore A' is B-free. We have a solution for the Row Deletion(B) instance with k row deletions.

"$\Leftarrow$" Assume that by deleting at most k rows in A we can make A B-free. Note that we cannot destroy any of the induced B in A by deleting at most k of the bottom $r'' + k$ rows of A. Therefore, it must be possible to delete at most k of the top n rows of A to make A B-free. Furthermore, from each induced B in A, at least one row must have been deleted. Thus, choosing the elements in $\mathcal{S}$ that correspond to the deleted rows into a set $\mathcal{S}'$ yields a solution of size k to the original r'-Hitting Set problem. □

The idea of the above proof—using the submatrix V of B_σ to encode an instance of d-Hitting Set—is employed in all of the following proofs. In order to show the "1:1-correspondence" of the original d-Hitting Set problem and the generated Row Deletion(B) instance, mainly two conditions need to be fulfilled:

> **Condition (1):** If the optimal solution to the d-Hitting Set instance has size k, there are no "cheaper" solutions for the generated Row Deletion(B) instance.
>
> **Condition (2):** If there is a solution of size k to the original d-Hitting Set instance, deleting the corresponding rows in A destroys all occurrences of B in A.

Whilst Condition (1) is rather straightforward to meet by extending the idea of the above proof, Condition (2) is quite intricate to fulfill in general, because it must be ensured that parts of Bs encoded into A due to different sets in $\mathcal{C}$ do not induce additional occurrences of B.

Proof (Theorem 4). Given an instance of r-Hitting Set and a σ-decomposition for the forbidden submatrix B, the resulting matrix A of this proof's reduction is composed of four submatrices: The upper left submatrix is generated by the encoding scheme presented in the proof of Theorem 3 using V as the forbidden submatrix, the upper right and lower left submatrices are filled with σ's. Two cases are distinguished for writing U into the lower right submatrix of A: If U does not induce V (Case I), the lower right submatrix has size $(k{+}1)r'' \times (k{+}1)s''$ and contains $k + 1$ times the matrix U in a diagonal scheme. If U induces V (Case II), the lower right submatrix has size $(k + 1)r'' \times s''$ and contains $k + 1$ copies of U—the two cases are illustrated by Fig. 3.

Observe that the reduction for Case I keeps the right part of A V-free. Recall that a single U by itself cannot induce V according to the prerequisite of Case I. However, if we would encode occurrences of U one upon the other as for Case II, on the one hand, V could be induced by rows from different encodings of U in the lower right part of A. On the other hand, U could be induced by several encodings of V in the upper left part of A. Consequently, we would

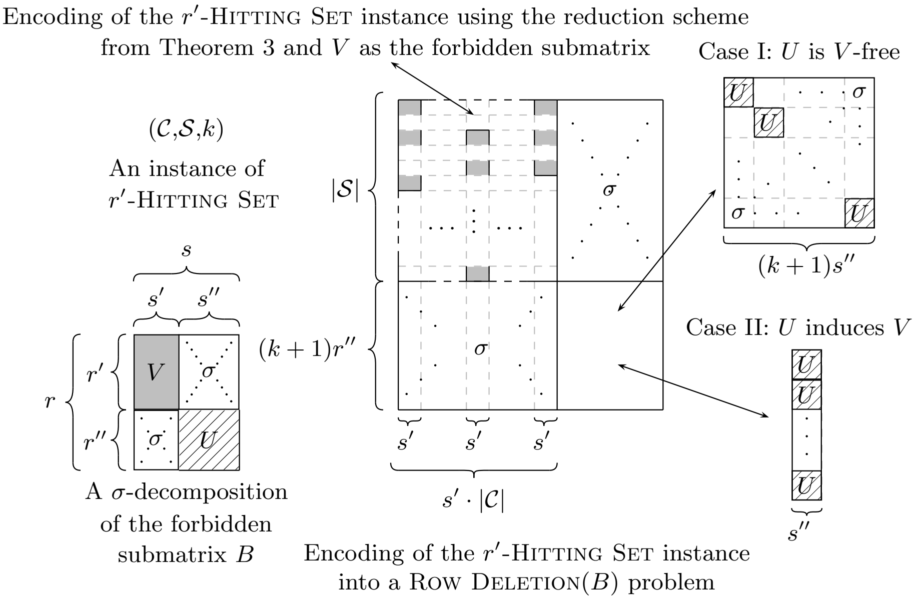

Encoding of the r'-Hitting Set instance using the reduction scheme from Theorem 3 and V as the forbidden submatrix
Case I: U is V-free
($\mathcal{C}$,$\mathcal{S}$,k)
An instance of r'-Hitting Set
$|\mathcal{S}|$
σ
U
$(k+1)s''$
s
s'
s''
r'
V
r
r''
$(k+1)r''$
Case II: U induces V
s'
s'
s'
$s' \cdot |\mathcal{C}|$
s''
A σ-decomposition of the forbidden submatrix B
Encoding of the r'-Hitting Set instance into a Row Deletion(B) problem

Fig. 3. Illustration of the reduction in the proof of Theorem 4.

have unwanted occurrences of B. The diagonal scheme avoids these unwanted occurrences of V and keeps in particular the right part of A B-free. In Case II, the reduction cannot keep the right part of A V-free because already a single occurrence of U induces V. However, B cannot be induced there because the reduction does not provide enough columns for an occurrence of B. Therefore, if we destroy all induced Vs in the upper left part of A , then U cannot be induced in the left part of A due to the prerequisite of Case II. Then, the matrix A can be made B-free, even if there are some occurrences of V in the right part of A.

Now, we show the "1:1-correspondence" of the solutions:

"⇒" Assume that we have a solution $\mathcal{S}'$ to the encoded r'-Hitting Set instance. Then, delete those of the n topmost rows in A that correspond to the elements in $\mathcal{S}'$, obtaining A'. Note that this destroys all occurrences of V in the left part of A (ensuring Condition (2)). For Case I, this directly implies that A' is V-free and therefore B-free. For Case II, recall that B is not induced in the right part of A since we can find no sufficiently large set of columns such that both U and V are induced there. But recall that since V is not induced in the left part of A', this also means that U is not induced there. Hence, A' is B-free.

"⇐" Assume that by deleting k rows, we can make A B-free. Note that by deleting one of the $(k+1)r''$ bottommost rows in A, we can destroy at most one induced B in A. Therefore, there is an optimal solution to Row Deletion(B) that involves only the deletion of k of the n topmost rows of A. The rest of the argument follows from the proof of Theorem 3: If, by deleting k of the n topmost rows of A, we can make A B-free, then from each V encoded into the

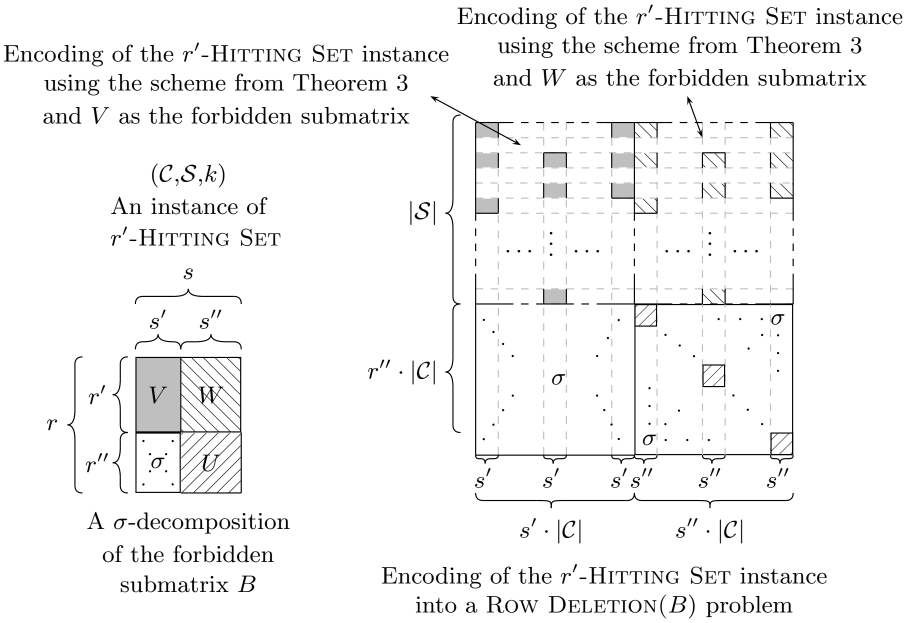

Encoding of the r'-Hitting Set instance using the scheme from Theorem 3 and W as the forbidden submatrix
Encoding of the r'-Hitting Set instance using the scheme from Theorem 3 and V as the forbidden submatrix
$(\mathcal{C},\mathcal{S},k)$
An instance of r'-Hitting Set
s
s'
s''
r
r'
r''
V
W
σ
U
A σ-decomposition of the forbidden submatrix B
$|\mathcal{S}|$
$r'' \cdot |\mathcal{C}|$
σ
s'
s'
s'
s''
s''
s''
$s' \cdot |\mathcal{C}|$
$s'' \cdot |\mathcal{C}|$
Encoding of the r'-Hitting Set instance into a Row Deletion(B) problem

Fig. 4. Reduction from r'-Hitting Set to Row Deletion(B) used in the proof of Theorem 1.

left part of A, at least one row must have been deleted (ensuring Condition (1)). Since each encoded V corresponds directly to a set in $\mathcal{C}$, choosing the elements in $\mathcal{S}$ that correspond to the deleted rows in A yields a solution to the original r'-Hitting Set instance. □

Proof (Theorem 1). [Sketch] As illustrated in Fig. 4, this reduction is similar to the one used in the proof for Case I of Theorem 4. The resulting matrix A is again composed of four submatrices, the reduction only differs in the construction of the upper right submatrix of A. In the upper right submatrix of A, the given r'-Hitting Set instance is encoded using the scheme from the proof of Theorem 3 and W as the forbidden submatrix. As in the previous two proofs, the parameter k is preserved and the encoding can be carried out in polynomial time with respect to the input size.

As in the proof of Theorem 4, the encoding process ensures that B is not induced in the right part of A. This is due to the following observation: Assume that B is induced in the right part of A. Then, V is induced there as well. By the prerequisites of the theorem, the non-encoding part of B does not induce V. Therefore, an occurrence of V involves columns from at least two different encodings of U. However, note that any two such columns of A can—due to the encoding scheme—only have less than r' rows that do not contain a σ. But V has r' rows with no symbol equal to σ, a contradiction.

Now note that by deleting one of the $r'' \cdot |\mathcal{C}|$ bottom rows in A, we can destroy at most one induced submatrix B in A—this could always be achieved by deleting one of the $|\mathcal{S}|$ topmost rows of A. Therefore, every solution of the produced Row Deletion(B) instance can be translated to one deleting only rows from the topmost $|\mathcal{S}|$ rows.

From the second observation it is clear that an optimal solution of Row Deletion(B) can only consist of the n topmost rows of A. The elements of $\mathcal{S}$ that correspond to the deleted rows form a solution for the original r'-Hitting Set instance, using the same argument as in the previous proofs. Conversely, if we have a solution of size at most k to the original r'-Hitting Set instance, deleting the corresponding rows in A destroys all occurrences of V in A due to the first observation and makes A B-free. Hence, every solution to the r'-Hitting Set instance $(\mathcal{S}, \mathcal{C}, k)$ implies a solution to the Row Deletion(B) instance (A, k) and vice versa. □

The scheme and ideas of the above proof are also used in proving Theorem 2. The details of the proof are rather involved, firstly establishing the result for $r \times 2$ matrices and then extending this result to obtain Theorem 2. We shall only present the main idea for the reduction involved, referring to [11] for details.

Proof (Theorem 2). [Reduction scheme for $r \times 2$ matrices, key ideas] For $r \times 2$ matrices that fulfill the prerequisites of the theorem, the reduction is performed as follows: Given an instance of r'-Hitting Set, the matrix A is constructed just as in the proof of Theorem 1. Then, the following algorithm is performed: As long as there are two columns in the right part of A whose upper n entries are identical to each other, remove one of the two columns from A. It is possible to show that after this "merging" of columns in A, the right part of A is B-free.

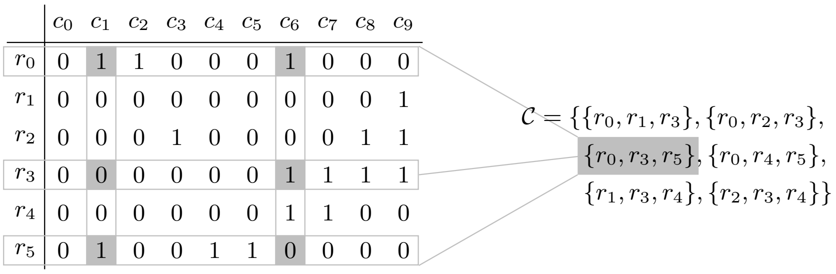
An extension of this scheme from $r \times 2$ matrices to all matrices that fulfill the prerequisites of this theorem is performed by arguing that by being able to avoid the occurrence of an $r \times 2$ submatrix of B in the right part of A, we are also able to avoid the occurrence of B there altogether. □

4 Algorithmic Tractability

This section points out an algorithmic approach to solve Row Deletion(B). To this end, we give a parameterized reduction to r-Hitting Set where r is the number of rows in B.

Theorem 5. *Given a fixed $r \times s$ submatrix B,* Row Deletion(B) *is parameter-preserving reducible to r-*Hitting Set *in $O(n^r \cdot m \cdot s \cdot r!)$ time.*

Proof. Given an instance (A, k) of Row Deletion(B) (where A is an $n \times m$ matrix), we construct an instance $(\mathcal{S}, \mathcal{C}, k)$ of r-Hitting Set as follows: (1) We construct a set $\mathcal{S} = \{r_0, \ldots, r_{n-1}\}$ containing one element for every row in A. (2) We compute a set $\mathcal{C}$ of subsets of $\mathcal{S}$: For every set $R \subseteq \{r_0, \ldots, r_{n-1}\}$ with $|R| = r$ corresponding to r rows in A that induce B, we add R to $\mathcal{C}$. (3) The



	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
r_0	0	1	1	0	0	0	1	0	0	0
r_1	0	0	0	0	0	0	0	0	0	1
r_2	0	0	0	1	0	0	0	0	1	1
r_3	0	0	0	0	0	0	1	1	1	1
r_4	0	0	0	0	0	0	1	1	0	0
r_5	0	1	0	0	1	1	0	0	0	0

$\mathcal{C} = \{\{r_0, r_1, r_3\}, \{r_0, r_2, r_3\},$
$\{r_0, r_3, r_5\}, \{r_0, r_4, r_5\},$
$\{r_1, r_3, r_4\}, \{r_2, r_3, r_4\}\}$

Fig. 5. Illustrating the reduction from Row Deletion(B) to r-Hitting Set. Let B be the matrix consisting of rows $(1, 1)$, $(0, 1)$, and $(1, 0)$. Given an input matrix A as shown, we generate an r-Hitting Set instance consisting of a set $\mathcal{S} = \{r_0, r_1, \ldots, r_5\}$ and the set $\mathcal{C}$ as shown. The gray underlay shows, as an example, how B is induced by rows r_0, r_3, and r_5 in columns c_1 and c_6 of A, leading to the the subset $\{r_0, r_3, r_5\}$ in $\mathcal{C}$.

parameter k is directly preserved. Then, $(\mathcal{S}, \mathcal{C}, k)$ is the resulting r-Hitting Set instance. An example for the reduction is illustrated in Fig. 5.

The direct "1:1-correspondence" between the solutions of the r-Hitting Set instance and the Row Deletion(B) instance can be shown as follows: Let $\mathcal{S}' \subseteq \mathcal{S}$ be a solution of size k to the r-Hitting Set instance $(\mathcal{S}, \mathcal{C}, k)$. We delete the rows in A that correspond to the elements in $\mathcal{S}'$, yielding A'. Assume that B were still induced in A' by a set I of rows. Then, the rows in I did induce B in A, meaning a set containing the elements corresponding to these rows was put into $\mathcal{C}$. But one row of I must then have been deleted since $\mathcal{S}'$ is a valid solution to $(\mathcal{S}, \mathcal{C}, k)$, a contradiction. Therefore, B cannot be induced by A' anymore.

If, on the other hand, A can be made B-free by deleting k rows, then for each occurrence of B in A by some rows, at least one of these rows must have been deleted. By choosing the elements corresponding to the deleted rows as a solution $\mathcal{S}' \subseteq \mathcal{S}$ to the generated r-Hitting Set instance, we have chosen at least one element from every set in $\mathcal{C}$, making $\mathcal{S}'$ a valid solution of size k.

The running time follows with Proposition 1. □

Theorem 5 directly implies the following two positive results:

- The best known polynomial-time approximation algorithm for r-Hitting Set, which currently has approximation factor r, can be used to obtain a factor-r approximation for the corresponding Row Deletion(B).
- r-Hitting Set can be trivially solved in $O(r^k \cdot n^r)$ time, where k denotes the size of the solution. This means that for constant r Row Deletion(B) is fixed-parameter tractable with respect to parameter k. See [7] for the currently best fixed-parameter algorithms for r-Hitting Set—for instance, the best exponential term for 3-Hitting Set is known to be 2.27^k instead of only 3^k.

5 Conclusion

In this work, we have started a systematic study on complexity of and algorithms for Row Deletion(B). Among others, we were able to show NP-completeness for a number of natural cases of forbidden submatrices B. It remains open to generalize all special cases treated in this work, e.g., by proving or disproving the following conjecture: For every forbidden submatrix B with at least three rows, Row Deletion(B) is NP-complete.

Our work was partially motivated by constructing perfect phylogenies from binary matrices [8, 10]. For this special case, where we have to consider a forbidden submatrix B consisting of the rows $(1, 1)$, $(1, 0)$, and $(0, 1)$, our results yield that Row Deletion(B) is at least as hard as 2-Hitting Set (which is the same as the well-known Vertex Cover problem) and that it always can be solved by transforming it into an instance of 3-Hitting Set.

Note that there remains a "gap" between the results of this work: Let an $r \times s$ forbidden submatrix B have a σ-decomposition with height-r' submatrix V. Then, if $r > r'$, we showed, on the one hand, that in certain cases Row Deletion(B) is at least as hard to solve as r'-Hitting Set and, on the other hand, that it is not harder to solve than r-Hitting Set.

References

1. A. Brandstädt, V. B. Le, and J. P. Spinrad. *Graph Classes: A Survey.* SIAM Monographs on Discrete Mathematics and Applications, 1999.
2. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, 1979.
3. J. Gramm, J. Guo, F. Hüffner, and R. Niedermeier. Graph-modeled data clustering: fixed-parameter algorithms for clique generation. In *Proc. of 5th CIAC*, volume 2653 of LNCS, pages 108–119, Springer, 2003.
4. J. Gramm, J. Guo, F. Hüffner, and R. Niedermeier. Automated generation of search tree algorithms for graph modification problems. In *Proc. of 11th ESA*, volume 2832 of LNCS, pages 642–653, Springer, 2003.
5. B. Klinz, R. Rudolf, and G. J. Woeginger. Permuting matrices to avoid forbidden submatrices. *Discrete Applied Mathematics*, 60:223–248, 1995.
6. A. Natanzon, R. Shamir, and R. Sharan. Complexity classification of some edge modification problems. *Discrete Applied Mathematics*, 113:109–128, 2001.
7. R. Niedermeier and P. Rossmanith. An efficient fixed-parameter algorithm for 3-Hitting Set. *Journal of Discrete Algorithms*, 1:89–102, 2003.
8. I. Pe'er, R. Shamir, and R. Sharan. On the generality of phylogenies from incomplete directed characters. In *Proc. of 8th SWAT*, volume 2368 of LNCS, pages 358–367, Springer, 2002.
9. R. Shamir, R. Sharan, and D. Tsur. Cluster graph modification problems. In *Proc. of 28th WG*, volume 2573 of LNCS, pages 379–390, Springer, 2002.
10. R. Sharan. *Graph Modification Problems and their Applications to Genomic Research.* Ph.D. Thesis, School of Computer Science, Tel-Aviv University, 2002.
11. S. Wernicke. *On the Algorithmic Tractability of Single Nucleotide Polymorphism (SNP) Analysis and Related Problems.* Diploma Thesis, WSI für Informatik, Universität Tübingen, September 2003.